1. Functional Description

The SENSOR integrates an absolute pressure sensor and a temperature sensor. Since the pressure sensor is cross sensitive for temperature, every P readout relies on the most recent T measurement. Measurement performance, power consumption, and device behavior can be configured to fulfill the requirements of different use cases. Features include configurable conversion time, configurable oversampling, interrupts, and a 32-slot memory that can be used as FIFO or as a moving average filter.

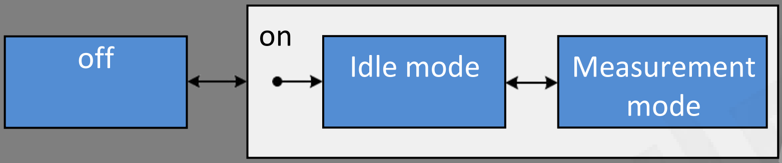

Figure 4.1: The system states

The device starts in idle mode, see Figure 4.1 for the system states. This mode is mainly used to configure the measurement process. Once in measurement mode the device performs pressure and temperature measurements as required; in order to save power it is possible to specify a standby time between two measurements.

Device states are selected by writing on MODE_CFG according to the following Table 4.1:

Table 4.1: Device states

CFG	State
0	Idle mode
1	Measurement mode

1.1 Operating mode description

Idle mode

Device configuration has to be performed in this operating mode; no measurements are performed. If HP bit is 0 (see MODE_CFG), the power consumption is low (I_{PD}) and only the following registers are accessible via SPI/I^{2}C: MODE_CFG, PART_ID, DATA_STAT, FIFO_STAT, PRESS_OUT, TEMP_OUT, INT_STAT.

By setting HP, the power consumption increases to I_{DE} and all registers become available.

Measurement mode

Operation in this mode depends from the value of STBY_T.

If STBY_T=0 the device performs continuous measurements (temperature and pressure).

If STBY_T=1 the device performs a single measurement (temperature and pressure) and goes into idle mode.

If STBY_T>1 the device alternates a measurement phase and a standby phase (pulsed operation); when HP=0 only a low power oscillator is active in the standby phase and the current draw is I_{PPD}.

The length of the standby phase can be selected from 10 ms to 600 s by writing on STBY_T. A timing diagram of pulsed operation is shown below in Figure 4.2.

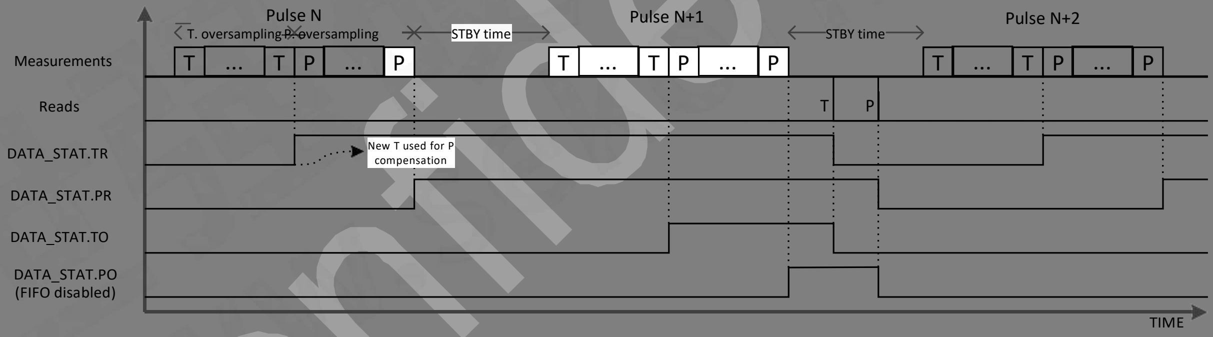

Figure 4.2: Timing diagram of measurement mode (PT_RATE=0)

It is possible to select a pressure/temperature rate by writing on PT_RATE; the value varies from 1 to 256. The device will perform a new temperature measurement once every PT_RATE pressure measurements (see Figure 4.3).

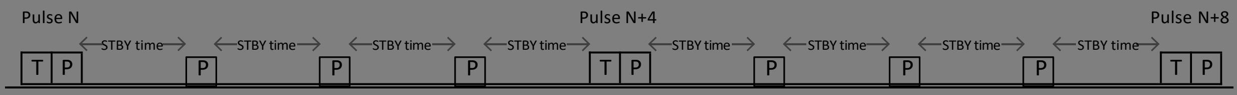

Figure 4.3: Effect of PT_RATE=1 (meaning P/T rate is 4)

MEAS_P and MEAS_T determine whether pressure or temperature measurements take place; in case only temperature or only pressure is enabled, PT_RATE has no effect. Depending from the value of OVST and OVSP, each measurement phase includes a certain number of single measurements used to calculate the average value.

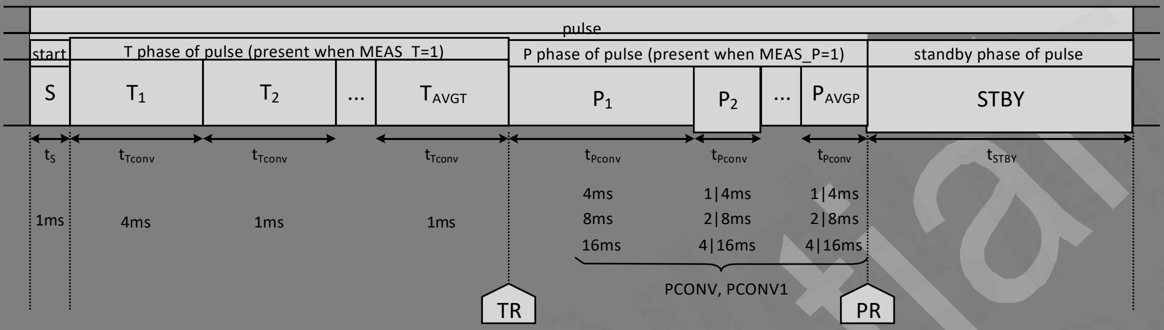

Figure 4.4: Duration of a pulse

Note: if the length of the standby phase is zero (continuous operation), only the first pressure measurement after a temperature measurement will be 4 times longer; the rest will maintain the nominal duration; the start-up phase is also not present in this case. When measuring only temperature, each oversampled measurement will start with a 4ms time followed by 1ms for the measurements required to complete over-sampling.

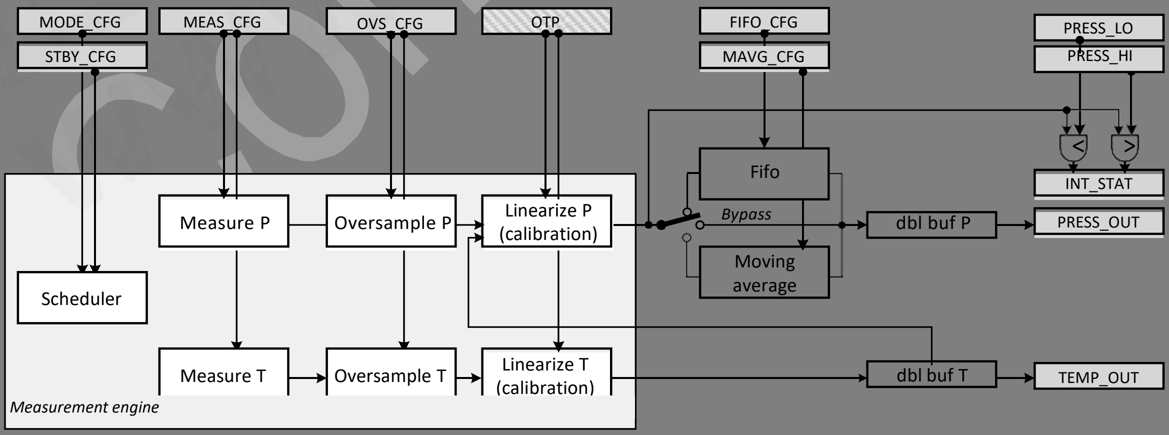

Figure 4.5: Pressure sensor data flow model

1.2 Temperature Sensor

The temperature sensor measures the junction temperature and outputs a calibrated value proportional to the absolute temperature.

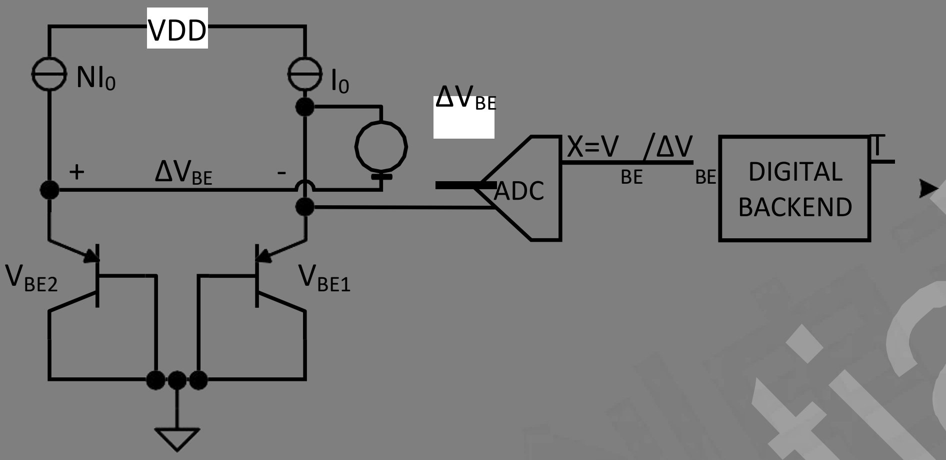

Figure 4.6: Temperature sensor structure

The junction temperature is measured using a high-precision ADC, as shown in Figure 4.6.The sensor accuracy is typically ± 0.2°C, with a conversion time of 4ms. Refer to Temperature Sensor Characteristics for detailed performance figures.

The temperature sensor is enabled by setting MEAS_T in MODE_CFG; it can be configured by writing on MEAS_CFG and OVS_CFG.

Oversampling (see OVST) is available to reduce the measurement noise at the expense of conversion time; between 1 and 128 measurements are performed and averaged to calculate the output value; the effect of oversampling on noise is shown in Table 3.7.

The temperature conversion time is 4ms; in case of oversampling, every additional conversion requires 1ms.

Sensor Readout

The temperature value is a 16 bit unsigned integer and is available in the sensor readout registers TEMP_OUT.

To ensure a consistent value during readout, TEMP_OUT registers are double buffered. When TEMP_OUT_L is read, the device copies all bytes from the internal measurement registers to the I^2C/SPI registers, then reads are always directly from the I^2C/SPI registers. Due to the implementation of this double buffering, it is not possible to guarantee the alignment between data ready flags and data if they are accessed in the same transition; it is advisable to access the flags separately from the data.

Temperature can be calculated in the Aquila V2.1 chip version as:

$T_{°C}$=TEMP_OUT/128 – 273.15

$T_{°K}$=TEMP_OUT/128

1.3 Absolute Pressure Sensor

The absolute pressure sensor determines the ambient pressure and outputs a calibrated and linearized value directly in Pa. Referring to the block diagram in Figure 1.1, the transducer C_X consists of a large-area membrane that acts as a capacitor. The capacitance change is proportional to the change in pressure. The capacitance is measured by a high-precision 3rd order sigma-delta converter.

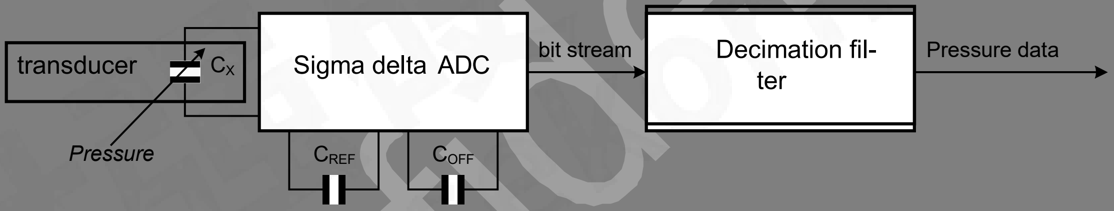

Figure 4.7: Absolute Pressure Sensor

The raw capacitance reading is automatically converted to absolute pressure through an optimized algorithm that compensates for the non-linearity of the transducer and for the temperature dependency, by combining calibration data stored in the internal OTP and the latest temperature measurement. It is advisable to perform a periodic temperature measurement in order to have an updated value; in order to correct for temperature variations a temperature measurement should be performed at least once per second. For fast temperature transients the temperature measurement should be performed more often to avoid errors in the temperature compensation. The exact requirements for temperature update rate depend on the use case and the physical properties of the soldering substrate, and should be evaluated by the user.

Configuration

The pressure sensor is enabled by setting MEAS_P in MODE_CFG; it can be configured by writing on MEAS_CFG, OVS_CFG, and MAVG_CFG.

Conversion time can be selected between 1ms, 2ms, 4ms; the first measurement after ADC power-up (i.e. after exiting standby mode or after a temperature measurement) requires 4 times the nominal duration.

Oversampling (see OVST) is available to reduce the measurement noise at the expense of conversion time; between 1 and 128 measurements are performed and averaged to calculate the output value.

The oversampled pressure value can either be transferred directly to the readout registers, can be recorded in a FIFO buffer, or can be further processed by a moving average filter; the desired path is selected by writing on MODE_CFG.FIFO_MODE.

1.4 Sensor Readout

The pressure value is a 24 bit unsigned integer and is available in the sensor readout registers PRESS_OUT and PRESS_OUT_F.

To ensure a consistent value during readout, PRESS_OUT registers are double buffered. When PRESS_OUT_XL is read, the device copies all bytes from the internal measurement registers to the I^2C registers, then reads are always directly from the I^2C registers. Due to the implementation of this double buffering, it is not possible to guarantee the alignment between data ready flags and data if they are accessed in the same transition; it is advisable to access the flags separately from the data.

Absolute pressure is represented as follows:

P_{Pa}=PRESS_OUT/64

NOTE: The PRESS_OUT value is already calibrated, linearized and temperature-compensated; it requires a division by 64 to achieve a pressure reading in Pa.

1.5 Special features

FIFO

A FIFO buffer with 32 positions is available to store pressure values; it is enabled by writing on MODE_CFG.FIFO_MODE.

The FIFO works as a circular buffer, with a write pointer used to store new pressure values and a read pointer used to retrieve them; the structure is shown in Figure 4.8. The write pointer is increased whenever a new measurement is complete, then the new value is written on the buffer; the pointer will increase even when the buffer is full, overwriting the oldest values recorded. Reads from PRESS_OUT_H increase the read pointer until it reaches the write pointer; subsequent reads will return 0. Both pointers wrap around (31 increments to 0).

Reading the entire FIFO quickly can be done by using PRESS_OUT_F; in this case a continuous read past address 0x29 will cycle the memory read pointer back to 0x27, increasing at the same time the FIFO read pointer. In this way the entire FIFO buffer can be transferred without addressing overhead.

NOTE: in order to read more than one value it is necessary to set the HP bit in MODE_CFG.

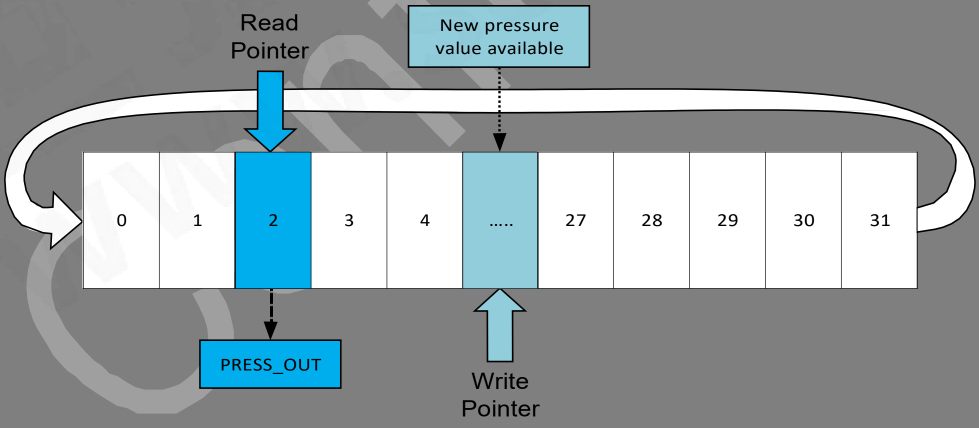

Figure 4.8: FIFO structure

FIFO operations are controlled by FIFO_CFG. The FIFO can be cleared by setting FP_CLEAR. FIFO status is available in FIFO_STAT; the current FIFO fill level (number of elements available) can be read on FP_FILL[4:0]. When FIFO is empty, FE is set; when FIFO is full, FF is set. If a new pressure value arrives while FIFO is full, the overrun bit DATA_STAT.PO is set, and the oldest value is overwritten.

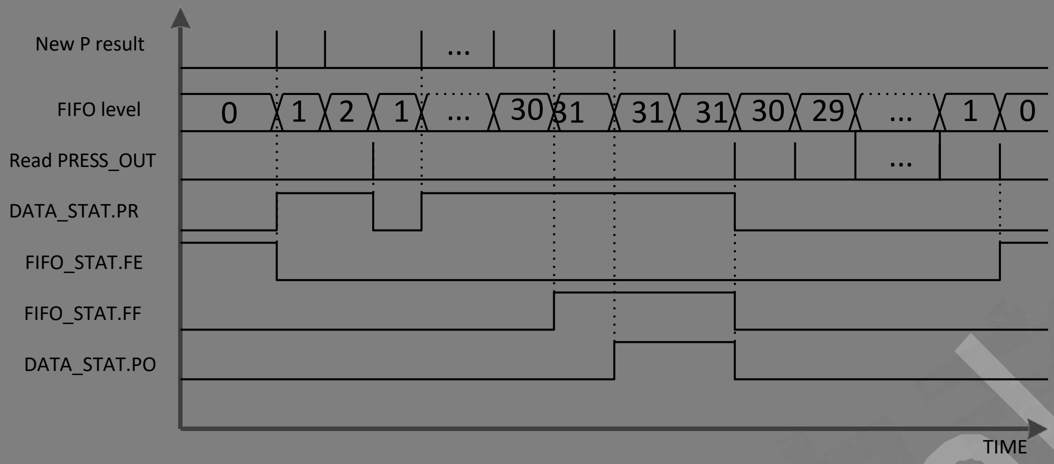

Figure 4.9: Sample FIFO timing diagram

Interrupts can be configured so that they will assert when FIFO is full, empty or filled to a certain level.

Interrupt

Interrupts can be enabled by writing on INT_CFG. Interrupts are available for the following events:

- Pressure measurement ready
- Temperature measurement ready
- FIFO level high
- FIFO full
- FIFO empty
- Pressure below PRESS_LO
- Pressure above PRESS_HI

Once an interrupt is asserted, the interrupt source can be read in INT_STAT.

Reading INT_STAT will automatically clear all interrupt source flags, will clear IA, and will reset the INT/SDOUT pin value to the de-asserted state.

The polarity of the INT pin can be selected by writing on INTF_CFG.

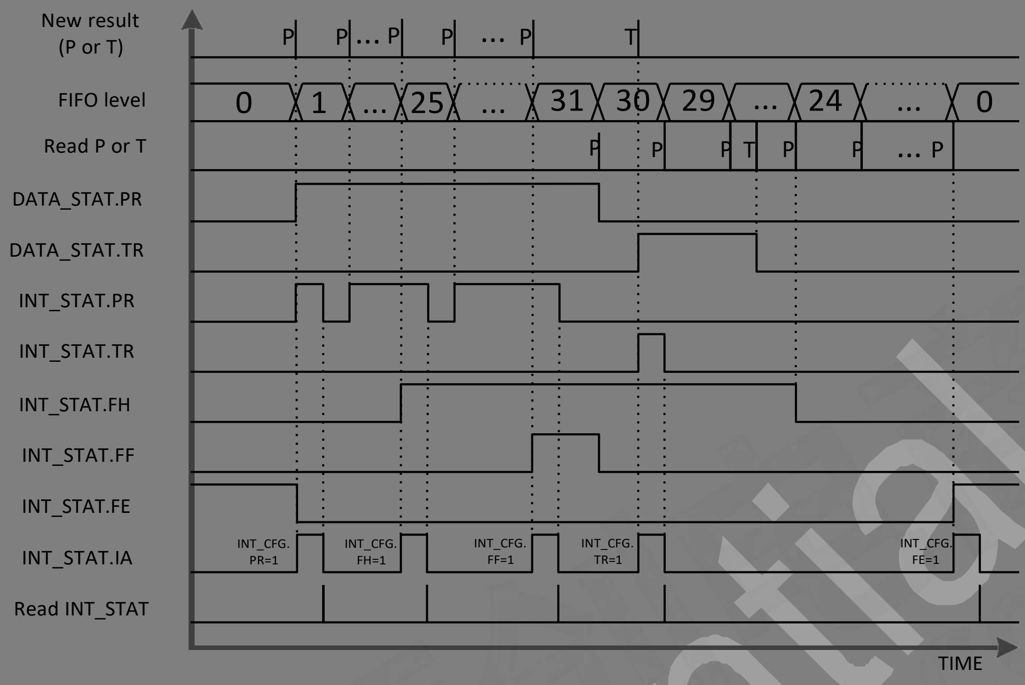

Figure 4.10: Sample interrupt timing diagram

Moving average

A moving average filter can be applied to pressure measurements. The filter is enabled by writing on MODE_CFG.FIFO_MODE. The FIFO buffer is used to implement this function, so FIFO is not functional when moving average is enabled.

The window size, i.e. the number of samples used by the moving average filter, is controlled by MAVG[2:0] (1 to 32); it should only be changed while the moving average filter is disabled. After moving average is enabled (by setting MAVG_EN), the first pressure value available is used to initialize the window with that first pressure value.

2. Interface selection

Selection between I^{2}C and SPI is done through CSN.

If CSN is high, the I^{2}C interface will be active and the SPI interface inactive. For systems where the host communicates via I^{2}C to this device, connect CSN to VDD permanently.

A falling edge of CSN will disable the I^{2}C interface until the next power-on cycle or software reset. The SPI host should make sure to generate a low pulse on CSN before communicating to other devices on the bus, in order to avoid that the device interprets SPI bus traffic as I^{2}C.

2.1 I^{2}C Interface

SENSOR is compliant to the I^{2}C bus specifications:

[UM10204, I²C-bus specification and user manual, Rev. 6, 4 April 2014].

The SENSOR is an I^{2}C slave device. The I^{2}C interface supports fast (400kbps) and high-speed (3400kbps) mode.

The device applies all mandatory I^{2}C protocol features for slaves: START, STOP, Acknowledge, 7-bit slave address, and also the optional clock stretching. The latter means that the master must support clock stretching in order to successfully communicate with the SENSOR.

None of the other optional features (10-bit slave address, General Call, Software reset, or Device ID) are supported, nor are the master features (Synchronization, Arbitration, START byte).

I^{2}C timings can be found within the electrical characteristics.

I2C Operations on registers

The SENSOR uses a register model to interact with the host. This means that an I^{2}C master can write a value to one of the registers of a slave, or that it can read from one of the registers of the slave. In the SENSOR, registers are addressed using 1 byte. The values stored in a register are also 1 byte. However, the SENSOR implements "auto increment" which means that it is possible to read, for example, two bytes by supplying the address of the first byte and then reading two bytes. The most significant bit of the address is used as disable for auto incrementing the register address; this can be useful for polling the same register.

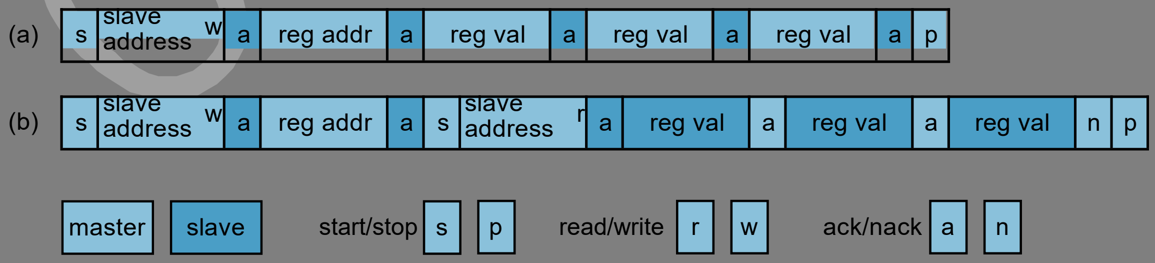

Figure 5.1: I2C Transaction Formats: a) write, b) read

A typical *write* transaction (see Figure 5.1a) therefore has the following format. The master initiates a transaction with a so-called start condition “s”. This blocks the bus. Next, the master sends the 7 bits SENSOR *slave address* followed by a 1 bit direction (a 0 indicating write “w”). This byte is acknowledged “a” by the slave. The master continues by sending the 8 bit *register address*, which is acknowledged by the slave. The remaining 7 LSB are stored in an internal CRA register (“Current Register Ad- dress”). Finally, the master sends the 8 bit *register value*, which is acknowledged by the slave (or nack’ed when the address is not writable). This value is written to the register pointed to by the CRA; the CRA is then incremented by 1 if auto increment is enabled. Optionally, the master sends more 8 bit values, for the next registers (auto incrementing CRA), each of which is (n)ack’ed by the slave. Finally, the master gen- erates a stop condition “p”, unblocking the bus for other transactions.

A *read* transaction (see Figure 5.1b) starts with a write (of the register address), fol- lowed by a read. Consequently, it has the following format. The master initiates the transaction with a start condition. Next, the master sends the 7 bits SENSOR *slave address* followed by a 1 bit direction (a 0 indicating write). This byte is acknowledged by the slave. The master continues by sending the 8 bit *register address* (composed as described in the write transaction) which is acknowledged by the slave and stored in the CRA register. Then the master sends another start condition (a so-called re- peated start condition, keeping the bus blocked) followed by the 7 bits SENSOR *slave address* followed by a 1 bit direction (a 1 indicating read “r”), which is acknowl- edged by the slave. Next, the slave sends an 8 bits *register value* from the register pointed to by the CRA register, and the CRA is incremented by 1 if auto increment is enabled. This byte is acknowledged by the master. The master may read another 8 bits (auto increment feature) from the slave and acknowledge that, until the master sends a nack “n” followed by a stop to unblock the bus.

The SENSOR has a 7 bit address space, potentially addressing 128 registers. In reality, only few addresses are actually backed by a register (see section Register Overview). All other addresses are *reserved*. A write transaction to a reserved (or read-only) register causes a not-acknowledge. A read transaction for a reserved register will return a 0.

I2C Slave Address

The SENSOR is an I^2C slave device with a slave address selectable at produc- tion time between 0x20 and 0x3F. For 0x20 this means that the first byte after a start condition is 0100 000x, where x indicates the data direction, so 0x40 (0100

0000) for write and 0x41 (0100 0001) for read. In case of 0x3F the first byte after a start condition is 0x7E (0111 1110) for write and 0x7F (0111 1111) for read.

High-speed mode

The bus operation speed is limited to 400kHz unless a high speed enable command (00001xxx) is issued by the master device as the first byte after START condition. This switches the bus to a high speed operation which allows data transfer frequencies up to 3.4MHz. Such command is not acknowledged by the slave but the input filter time constant on the serial interface (SDA and SCL) is adapted to allow higher transfer rate. After a high-speed command, the slave address is transmitted by the master in order to invoke a data transfer. The bus keeps operating at the highest operating frequency until the master issues an STOP condition. Upon reception of the STOP condition by the slave, the input filters are switched to their initial time constants, which allow only up to 400kHz transfer rates.

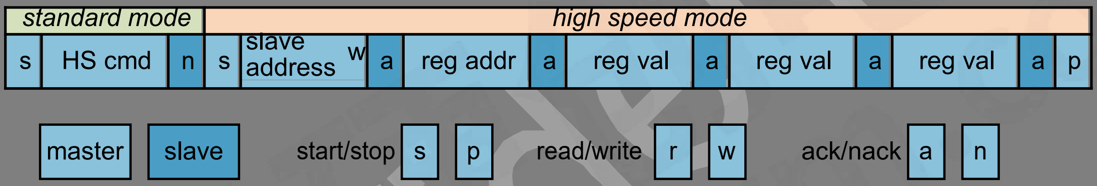

Figure 5.2: High-speed mode selection

2.2 SPI Interface

The SENSOR is an SPI bus slave. The SPI allows to write and read the registers of the device. The serial interface interacts with the outside world with 4 wires: CSN, SCLK, SDIN and SDOUT; an optional 3 wire mode uses a single bi-directional data line in- stead of two.

CSN is the serial port enable and it is controlled by the SPI master. It is driven low at the start of the SPI frame and returns high at the end.

SCLK is the serial port clock and is controlled by the SPI master; it should stay high in the absence of transmissions. SDIN and SDOUT are respectively the serial port data input and output. Those lines are driven at the falling edge of SCLK and should be captured at the rising edge of SCLK.

The SDOUT pin is shared between SPI output data and interrupt function. When CSN=1 and interrupt functionality is enabled, SDOUT will provide interrupt service.

If interrupts are disabled then SDOUT stays in high impedance mode until data is requested from the device, i.e. during an SPI transaction.

Both the read register and write register commands are completed in 16 clock pulses or in multiples of 8 in the case of multiple byte read/write. Bit duration is the time between two falling edges of SCLK. The first bit (bit 0) starts at the first falling edge of SCLK after the falling edge of CSN while the last bit (bit 15, bit 23, ...) starts at the last falling edge of SCLK just before the rising edge of CSN.

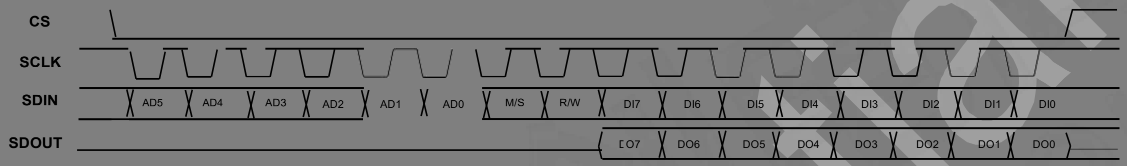

Figure 5.3: Minimum SPI frame

The device can also work with a low idle value of SCLK (also known as SPI mode 0), as shown below.

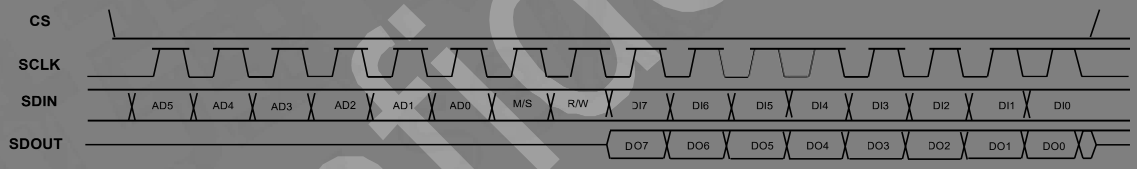

Figure 5.4: SPI frame using mode 0

A standard SPI frame is organized as follows:

Table 5.1: Description of an SPI frame

Bit	Name	Description
Bit 0-5	AD[5:0]	Address of the indexed register
Bit 6	MS	0: address will be auto incremented in multiple read/write commands. 1: address will remain unchanged in multiple read/write commands.

Bit	Name	Description
Bit 7	RW	0: data DI(7:0) is written into the device. 1: data DO(7:0) is read from the device. In the latter case, the chip will drive SDOUT at the start of bit 8.
Bit 8-15	DI(7:0)	Data written to the device (MS bit first)
	DO(7:0)	Data written to the device (MS bit first)

In multiple read/write commands, further blocks of 8 clock periods are added. When the MS bit is 0 the address used to read/write data remains the same for every block. When MS bit is 1 the address used to read/write data is increased at every block. The function and the behavior of SDIN and SDOUT remain unchanged. SPI timings can be found within the electrical characteristics.

SPI read

The SPI single byte read command consists of 16 clock pulses. A multiple byte read is performed by adding blocks of 8 clock pulses to the single byte read. All register data is copied into an intermediate buffer at the beginning of a read transaction; multiple reads in the same read transaction will refer to the intermediate buffer.

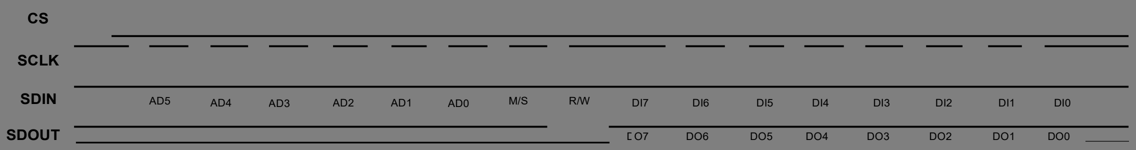

Figure 5.5: SPI read timing diagram

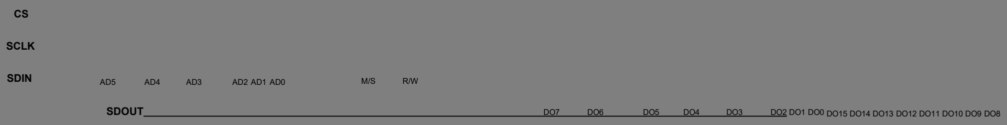

Figure 5.6: Multiple byte SPI read (2 bytes)

SPI write

The SPI single byte write command consists of 16 clock pulses. A multiple byte write is performed by adding blocks of 8 clock pulses to the single byte write.

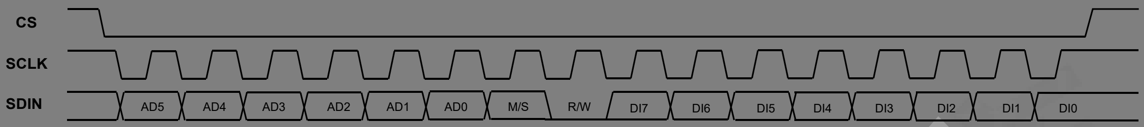

Figure 5.7: Single byte write

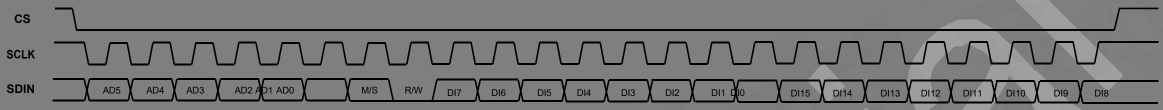

Figure 5.8: Multiple byte write (2 bytes)

SPI read in 3-wire mode

A 3-wire SPI mode can be selected by setting bit SPI3 in the INTF_CFG register. In this case, SDA acts as a bi-directional data line, and should be connected to an external pull-up resistor. Multiple byte read is also available in 3-wire mode.

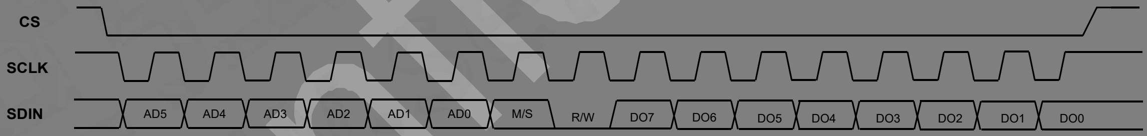

Figure 5.9: SPI read in 3-wire mode

3. Register Description

This section describes the registers of SENSOR.

Register Overview

Address	Name	Type	access in LP mode	Default	Description
0x00	PART_ID0	R	Y	0x2X	Part ID [7:0]
0x01	PART_ID1	R	Y	0x03	Part ID [15:8]
0x02	UID0	R	N		Unit ID [7:0]
0x03	UID1	R	N		Unit ID [15:8]
0x04	UID2	R	N		Unit ID [23:16]
0x05	UID3	R	N		Unit ID [31:24]
0x06	MODE_CFG	R/W	Y	0x00	Device configuration
0x07	MEAS_CFG	R/W	N	0x08	Measurement configuration, conversion time
0x08	STBY_CFG	R/W	Y (W)[3]	0x00	Standby time configuration
0x09	OVS_CFG	R/W	N	0x00	Oversampling setting for pres-sure and temperature
0x0A	MAVG_CFG	R/W	N	0x00	Moving average configuration
0x0B	INTF_CFG	R/W	Y (W)[3]	0x00	Interface configuration
0x0C	INT_CFG	R/W	Y (W)[3]	0x7F	Interrupt mask configuration
0x0D	PRESS_LO_XL	R/W	N	0x00	Low pressure threshold [7:0]
0x0E	PRESS_LO_L	R/W	N	0x00	Low pressure threshold [15:8]
0x0F	PRESS_LO_H	R/W	N	0x00	Low pressure threshold [23:16]
0x10	PRESS_HI_XL	R/W	N	0xFF	High pressure threshold [7:0]
0x11	PRESS_HI_L	R/W	N	0xFF	High pressure threshold [15:8]
0x12	PRESS_HI_H	R/W	N	0xFF	High pressure threshold [23:16]
0x13	FIFO_CFG	R/W	N	0x00	FIFO configuration
0x14	DATA_STAT	R	Y	0x00	Measurement data status
0x15	FIFO_STAT	R	N	0x02	FIFO status
0x16	INT_STAT	R	Y	0x08	Interrupt status
0x17	PRESS_OUT_XL	R	Y[4]	0x00	Pressure value [7:0]
0x18	PRESS_OUT_L	R	Y[4]	0x00	Pressure value [15:8]
0x19	PRESS_OUT_H	R	Y[4]	0x00	Pressure value [23:16]
0x1A	TEMP_OUT_L	R	Y	0x00	Temperature value [7:0]
0x1B	TEMP_OUT_H	R	Y	0x00	Temperature value [15:8]
0x27	PRESS_OUT_F_XL	R	Y[4]	0x00	Pressure value [7:0]
0x28	PRESS_OUT_F_L	R	Y[4]	0x00	Pressure value [15:8]
0x29	PRESS_OUT_F_H	R	Y[4]	0x00	Pressure value [23:16]

[3] Write access only
[4] Only the latest value; see description for detailed info.

3.1 Detailed register description

PART_ID[1-0] (address 0x00-0x01)

Table 6.1: Part identifier. Default value: 0x032X

PART_ID[15:0]			
PART_ID1		PART_ID0	
BIT15	BIT8	BIT7	BIT0

PART_ID[3:0] are defined in the OTP

UID[3:0] (address 0x02-0x05)

Table 6.2: 32 Bit unique device identifier

UID[31:0]			
UID3	UID2	UID1	UID0
BIT31	BIT16	BIT15	BIT0

MODE_CFG (address 0x06)

Table 6.3: Mode configuration register. Default value: 0x00

HP	FIFO_MODE[1:0]		CFG	RESET	X	MEAS_T	MEAS_P
BIT7							BIT0

Bit 7: HP: high power.

1: All registers are accessible via SPI/I^2C; power consumption is high. A delay of T_{pup} is required from high power enable to the next SPI/I^2C access.

0: The following registers are accessible via SPI/I^2C: MODE_CFG, PART_ID[1:0], DATA_STAT, FIFO_STAT, PRESS_OUT, TEMP_OUT, INT_STAT. Power consumption is low.

Table 6.4: Bit 6-5: FIFO_MODE[1:0]: configuration of pressure data path.

FIFO_MODE[1:0]	Pressure data path
0	Direct path
1	FIFO
2	Moving average

Table 6.5: Bit 4: CFG: operating mode configuration. Device states are selected according to this bit

CFG	State
1	Measurement mode
0	Idle mode

This bit is automatically reset in case of one-shot operation, after the required measurements have been performed

Bit 3: RESET: device reset

1: the device is reset to the power-on configuration. RESET is automatically cleared.

Table 6.5: Bit 1-0: MEAS_T, MEAS_P: measurement selection

MEAS_T	MEAS_P	
0	1	Only pressure measurements are enabled; the device will begin with a temperature measurement, then it will continue measuring only pressure
1	0	Only temperature measurement is enabled.
1	1	Pressure and temperature measurements are enabled. PT_RATE controls the temperature interleaving timer

MEAS_CFG (address 0x07)

Table 6.6: Measurement configuration register. Default value: 0x08

	X		P_CONV[1:0]		PT_RATE[2:0]		
BIT7							BIT0

Table 6.7: Bit 4-3: P_CONV[1:0]: pressure ADC conversion time.

P_CONV[1:0]	Conversion time [ms] First conversion	Conversion time [ms] Next conversions
0	4	1
1	8	2
2	16	4

Table 6.8: Bit 2-0: PT_RATE[2:0]: determines the ratio between P and T measurements as produced by the measurement engine (see Figure 4.5).

PT_RATE[2:0]	P/T rate
0	1
1	4
2	8
3	16
4	32
5	64
6	128
7	256

STBY_CFG (address 0x08)

Table6.9: Standby time configuration register. Default value: 0x00

X	STEP_T[2:0]			STBY_T[3:0]			
BIT7							BIT0

Table 6.10: Bit 3-0: STBY_T[3:0]: standby time. Measurements will be alternated with a standby phase with the following duration:

STBY_T[3:0]	Standby time (ms)
0	Continuous operation
1	One-shot operation (device returns to idle after one measurement is produced by the measurement engine)
2	10
3	20
4	30
5	50
6	100
7	250
8	500
9	750
10	1000
11	2000
12	5000

STBY_T[3:0]	Standby time (ms)
13	10000
14	60000
15	600000

Table 6.11: Bit 6-4: STEP_T[3:0]: step duration. In pulsed mode the converter current will be increased in 8 steps; STEP_T specifies the step duration:

STEP_T[2:0]	Step duration (ms)
0	No steps
1	0.12
2	0.24
3	0.48
4	0.97
5	1.9
6	3.9
7	7.7

OVS_CFG (address 0x09)

Table 6.12: Oversampling configuration register. Default value: 0x00

X	X	OVSP[2:0]			OVST[2:0]		
BIT7							BIT0

Oversampling works by averaging the value of a certain number of samples.
Oversampling applies to all measurement modes.

Table 6.13: Bit 5-3: OVSP[2:0]: oversampling of pressure measurements.

OVSP[2:0]	Number of averages
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Table 6.14: Bit 2-0: OVST[2:0]: oversampling of temperature measurements.

OVST[2:0]	Number of averages
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

MAVG_CFG (address 0x0A)

Table 6.15: Moving average configuration register. Default value: 0x00

X	X	X	X	X	MAVG[2:0]		
BIT7							BIT0

Table 6.16: Bit 2-0: MAVG[2:0], moving average configuration. Controls the number of samples used by the moving average filter.

MAVG[2:0]	Samples
0	1
1	2
2	4
3	8
4	16
5	32

INTF_CFG (address 0x0B)

Table 6.17: Host interface configuration register. Default value: 0x00

X	X	X	X	X	INT_EN	INT_HL	SPI3
BIT7							BIT0

Bit 2: INT_EN: interrupt enable

1: INT/SDOUT is controlled by INT_STAT.IA.

0: INT/SDOUT is always inactive (level of INT/SDOUT equals INT_HL).

Bit 1: INT_HL: interrupt polarity

1: INT/SDOUT low signals that interrupt is asserted.

0: INT/SDOUT high signals that interrupt is asserted.

Bit 0: SPI3: SPI mode control

1: SPI works in 3-wire mode

0: SPI works in 4-wire mode

INT_CFG (address 0x0C)

Table 6.18: Interrupt configuration register. Default value: 0x7F

X	TR	FH	FF	FE	PR	PH	PL
BIT7							BIT0

BIT 6	**TR**	Temperature data ready 1: interrupt is triggered when new temperature data becomes available. 0: interrupt disabled.
BIT 5	**FH**	Pressure FIFO level high 1: interrupt is triggered when the filling level of the pressure fifo becomes greater than the programmed level in FP_FILL_TH. 0: interrupt disabled.
BIT 4	**FF**	Pressure FIFO is full 1: interrupt is triggered when the pressure fifo becomes full (32 values). 0: interrupt disabled.
BIT 3	**FE**	Pressure FIFO is empty 1: interrupt is triggered when the pressure fifo becomes empty. 0: interrupt disabled.
BIT 2	**PA**	Pressure data is available 1: interrupt is triggered when new pressure data becomes available.

		0: interrupt disabled.
BIT 1	**PH**	Pressure is high
		1: interrupt is triggered when the pressure is greater than the programmed pressure high threshold (PRESS_HI).
		0: interrupt disabled.
BIT 0	**PL**	Pressure is low
		1: interrupt is triggered when the pressure is lower than the programmed pressure low threshold (PRESS_LO).
		0: interrupt disabled.

PRESS_LO (address 0x0D-0x0F)

Table 6.19: Low-pressure threshold. Default value: 0x000000

PRESS_LO[23:0]					
PRESS_LO_H		PRESS_LO_L		PRESS_LO_XL	
BIT23	BIT16	BIT15	BIT8	BIT7	BIT0

This register sets the pressure threshold for the low pressure interrupt.

PRESS_HI (address 0x10-0x12)

Table 6.20: High pressure threshold. Default value: 0xFFFFFF

PRESS_HI[23:0]					
PRESS_HI_H		PRESS_HI_L		PRESS_HI_XL	
BIT23	BIT16	BIT15	BIT8	BIT7	BIT0

This register sets the pressure threshold for the high pressure interrupt.

FIFO_CFG (address 0x13)

Table 6.21: FIFO configuration register. Default value: 0x00

X	X	FP_CLEAR	FP_FILL_TH[4:0]				
BIT7							BIT0

Bit 5: FP_CLEAR: FIFO clear.
1: the content of the FIFO is cleared; FP_CLEAR is automatically cleared.
0: no operation

Bit 4-0: FP_FILL_TH[4:0]: FIFO level threshold.
The value is used as target for interrupt and status generation.

DATA_STAT (address 0x14)

Table 6.22: Data status register. Default value: 0x00

X	X	X	X	PO	TO	PR	TR
BIT7							BIT0

Bit 3: PO: pressure overwrite.

With FIFO enabled (MODE_CFG.FIFO_MODE=1), this bit is set whenever a new pressure measurement is produced by the measurement engine while the pressure FIFO was already full. PO is cleared after reading PRESS_OUT_H or PRESS_OUT_F_H.

With FIFO disabled (direct path or moving average selected), this bit is set whenever a new pressure measurement is produced by the measurement engine and the previous data was not read. PO is cleared after reading PRESS_OUT_H or PRESS_OUT_F_H.

Bit 2: TO: temperature overwrite.

This bit is set when a new temperature measurement is produced by the measurement engine and the previous data was not read. TO is cleared after reading TEMP_OUT_H.

Bit 1: PR: pressure ready.

This bit is set when new pressure data becomes available. PR is cleared after reading PRESS_OUT_H.

Bit 0: TR: temperature ready.

This bit is set when new temperature data becomes available. TR is cleared after reading TEMP_OUT_H.

FIFO_STAT (address 0x15)

Table 6.23: FIFO status register. Default value: 0x02

FP_FILL[4:0]					FF	FE	FH
BIT7							BIT0

Bit 7-3: FP_FILL[4:0]: FIFO level.
Fill level of the pressure FIFO. FP_FILL is equal to 0x1F when FIFO has 31 but also 32 elements.

Bit 2: FF is set when the FIFO is enabled and is full (32 elements).

Bit 1: FE is set when the FIFO is enabled and is empty (0 elements).

Bit 0: FH is set when is enabled and the number of elements is greater than FP_FILL_TH.

INT_STAT (address 0x16)

Table 6.24: Interrupt status register. Default value: 0x08

IA	TR	FH	FF	FE	PR	PH	PL
BIT7							BIT0

Each bit except IA is set if the corresponding interrupt source has generated an event; reading this register will clear all flags.

Bit 7: IA: general interrupt flag; it is set if any of the interrupt flags (bits 6:0) relative to sources enabled in INT_CFG are set; see also INTF_CFG.INT_EN.

Bit 6: TR is set when a new temperature measurement is produced by the measurement engine.

Bit 5: FH is set when FP_FILL > FP_FILL_TH.

Bit 4: FF is set when the FIFO is full.

Bit 3: FE is set when FIFO is empty.

Bit 2: PR is set when new pressure measurement is produced (i.e. without considering oversampling; refer to Figure 4.5).

Bit 1: PH is set when the measurement engine produces a pressure measurement that is greater than PRESS_HI.

Bit 0: PL is set when the measurement engine produces a pressure measurement that is lower than PRESS _LO.

PRESS_OUT (address 0x17-0x19)

Table 6.25: Pressure value. Default value: 0x000000

PRESS_OUT[23:0]					
PRESS_OUT_H		PRESS_OUT_L		PRESS_OUT_XL	
BIT23	BIT16	BIT15	BIT8	BIT7	BIT0

This register contains a 24 bit integer (unsigned) representing the pressure in 1/64 Pa.

When the FIFO is enabled, a read on this register extracts one element from the fifo. When the FIFO is empty the read returns 0x000000.

When the FIFO is not enabled (bypass or moving average), reads from this register return the latest measurement result. If reads occur faster than measurements, values are repeated.

Due to the implementation of double buffering, it is not possible to guarantee the alignment between data ready flags and data if they are accessed in the same transition; it is advisable to access the flags separately from the data.

NOTE: HP must be 1 when reading this register with FIFO_MODE=1, in order to access the whole FIFO.

TEMP_OUT (address 0x1A-0x1B)

Table 6.26: Temperature value. Default value: 0x0000

TEMP_OUT[15:0]			
TEMP_OUT_H		TEMP_OUT_L	
BIT15	BIT8	BIT7	BIT0

This register contains an unsigned 16 bits integer representing the temperature as follows:

$T_{^\circ K}$=TEMP_OUT /128

Reads from this register return the latest measurement result. If reads occur faster than measurements, values are repeated. Due to the implementation of double buffering, it is not possible to guarantee the alignment between data ready flags and data if they are accessed in the same transition; it is advisable to access the flags separately from the data.

PRESS_OUT_F (address 0x27-0x29)

Table 6.27: Pressure value (fast FIFO read). Default value: 0x000000

PRESS_OUT[23:0]					
PRESS_OUT_F_H		PRESS_OUT_F_L		PRESS_OUT_F_XL	
BIT23	BIT16	BIT15	BIT8	BIT7	BIT0

This register is the same as PRESS_OUT, except that 0x29 auto increments to 0x27, so a single I2C/SPI transaction can be used to read multiple P measurements.

NOTE: HP must be 1 when reading this register with FIFO_MODE=1, in order to access the whole FIFO.